ASSOCIATION RULES

Olayların birlikte gerçekleşme durumlarını çözümleyen data mining yöntemleridir.

Support:

X->Y için hem X hem Y içeren transactionların tüm transactionlara oranı

Confidence:

X->Y için hem X hem Y içeren transactionların X içeren transactionlara oranı

Apriori Algorithm:

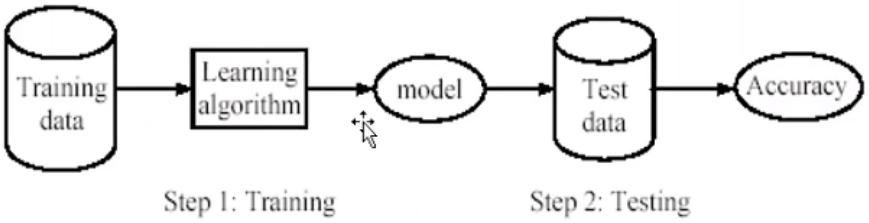
1. Min. support ve min. confidence değerlerini belirle
2. Min. support değerinin üstündeki itemları tek tek seç, diğerlerini ele
3. Min. support değerinin üstündeki 2-itemsetleri seç, diğerlerini ele
4. Min. support değerinin üstündeki 3-itemsetleri seç, diğerlerini ele
5. ….
6. Min. support değerinin üstündeki k-itemsetleri seç, diğerlerini ele
7. Min. support değerinin altına düşerse dur, önceki k-itemset'e kadarki kısım yeterli
8. Frequent itemsetler generate edilmiş oldu
9. Bu frequent itemsetin tüm subsetleri de frequent itemsettir
10. Subsetleri oluştur
11. Confidence değerlerini hesapla

Sequential Pattern Mining:

Sıralı olarak takip edilen patternlerden veri elde etme metodlarıdır.

SUPERVISED LEARNING (Has labels)

$$Accuracy = \frac{\text{Number of correct classifications}}{\text{Total number of test cases}},$$



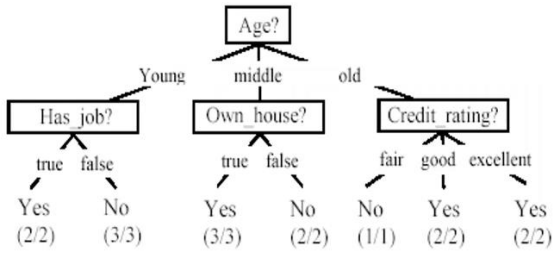
* Learning data ve test data birbirine yakın distributionlara sahip olmalı

Decision Tree Classification:

Approved or not

ID	Age	Has_Job	Own_House	Credit_Rating	Class
1	young	false	false	fair	No
2	young	false	false	good	No
3	young	true	false	good	Yes
4	young	true	true	fair	Yes
5	young	false	false	fair	No
6	middle	false	false	fair	No
7	middle	false	false	good	No
8	middle	true	true	good	Yes
9	middle	false	true	excellent	Yes
10	middle	false	true	excellent	Yes
11	old	false	true	excellent	Yes
12	old	false	true	good	Yes
13	old	true	false	good	Yes
14	old	true	false	excellent	Yes
15	old	false	false	fair	No

=>



Classification Using Association Rules:

* Örnek: own_house = true => class = yes [sup=6/15, conf=6/6]

Naive Bayesian Classification:

* Bayesian theorem: $P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})}$

$$P(C_1|\mathbf{X}) = \frac{P(\mathbf{X}|C_1)P(C_1)}{P(\mathbf{X})}$$

$$P(C_n|\mathbf{X}) = \frac{P(\mathbf{X}|C_n)P(C_n)}{P(\mathbf{X})}$$

Örnek:

- Bilgisayar alanların oranı = 0.643
- Bilgisayar almayanların oranı = 0.357
- Bilgisayar alanlar arasından yaşı 30'dan küçük olanların oranı = 0.222
- Bilgisayar almayanlar arasından yaşı 30'dan küçük olanların oranı = 0.6
- Bilgisayar alanlar arasından orta seviye gelire sahip olanların oranı = 0.444
- Bilgisayar almayanlar arasından orta seviye gelire sahip olanların oranı = 0.4

…

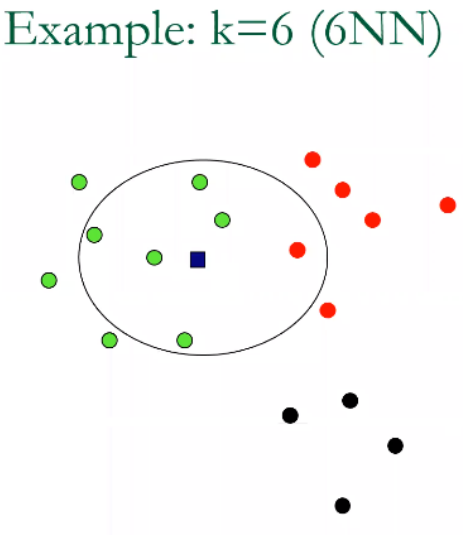
* Bilgisayar alınan tüm durumların olasılıklarını birbiriyle çarp, sonucu bilgisayar alma olasılığıyla çarp

* Bilgisayar alınmayan tüm durumların olasılıklarını birbiriyle çarp, sonucu bilgisayar almama olasılığıyla çarp

K-Nearest Neighborhood Algorithm:

* Veriyi model olarak kullanır

- Seçilen nokta ile tüm noktalar arasındaki mesafeleri tek tek hesapla
- Seçilen noktaya en yakın mesafedeki k (parametre) adet elemanı seç ve bunları P kümesine koy
- P kümesindeki elemanların class'ları arasından en büyük orana sahip olan class'ı sonuç olarak al



UNSUPERVISED LEARNING (Has not labels)

* Attribute, label yerine geçer

* Verileri ortak attribute'lara göre clusterlara ayırmaya dayanır

* Inter-cluster distance: max olmalı

* Intra-cluster distance: min olmalı

K-Means Clustering:

1. Kaç cluster olacağını (k'yi) seç
2. Rastgele k adet ayrı nokta seç, bu noktalar inital cluster
3. 1. noktanın tüm initial cluster'lara olan mesafesini hesapla
4. 1. noktayı kendisine en yakın inital cluster'a assign et
5. 2. noktanın tüm initial cluster'lara olan mesafesini hesapla
6. 2. noktayı kendisine en yakın inital cluster'a assign et
7. …
8. Tüm noktalar için bu işlemleri uyguladıktan sonra her bir cluster'ın mean değerini hesapla
9. Bulunan mean değerlerini inital cluster olarak kabul et ve tekrar tüm noktaların inital cluster'lara olan mesafelerini hesaplayıp kendine en yakın olan cluster'a assign et
10. Son iterasyonda hiçbir cluster'da değişiklik olmayana dek devam et
11. Eğer variance çok yüksek çıkarsa (bir cluster çok büyükken diğeri çok küçük gibi) tekrar dene, bu sefer seçilen rastgele noktalar daha düzgün sonuç verebilir

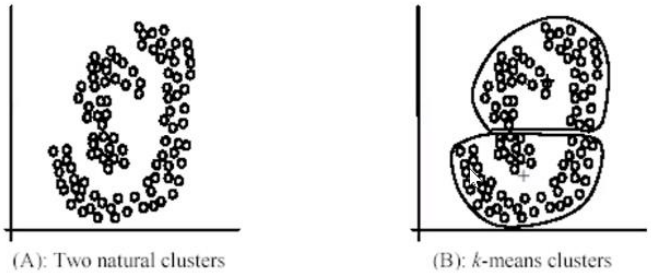
K'yi neye göre seçiyoruz? Tahmini, deneyerek (variance en azken en iyi)

<u>Avantajları:</u>

- Anlaması ve kodlaması kolay
- Verimli time complexity
- Cluster sayısı ve iterasyon sayısı düşük seçilirse lineer çalışır: O(n)
- En yaygın clustering algoritmasıdır

<u>Dezavantajları:</u>

- Yalnızca ortalama değeri hesaplanabilen veriler için uygulanabilir
- K değeri elle belirlenmek zorunda
- Initial seed çalışmasını engelliyor, doğru çalışması için tekrar çalıştırmak gerekebiliyor
- Outlier (çok uzak) değerleri de hesaba katar, bu da hatayı arttırır
    - Öncesinde outlier detection and removal algoritmaları kullanılarak önlenebilir
    - Veya algoritmayı verinin seçilen bir kısmı için uygulayarak önlenebilir
- Hyper-ellipsoid olmayan clusterlarda düzgün çalışmaz, örnek:



(A): Two natural clusters (B): *k*-means clusters

Cluster'ları nasıl ifade edebiliriz?

- Centroid (merkez) ve radius (yarıçap) ile
- Classification model with borders (2 < x < 5 ve 0 < y < 1 gibi)

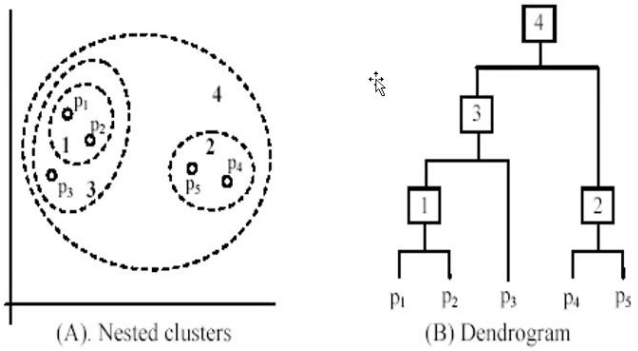
Hierarchical Clutering:

* Non-ellipsoid şekiller için daha iyi bir çözüm

* Dendogram (tree) kullanılır

Agglomerative (Bottom Up) Clustering: (Daha popüler)

- Her noktayı bir cluster kabul eder
- Tüm clusterlar arasından en yakın ikilileri bulur ve o ikilileri kendi içinde merge eder
- Tekrar en yakın ikilileri bulur ve merge eder
- …
- Tüm clusterlar en üstte tek bir cluster'a merge edilene kadar devam eder
- Tree'nin en üstünde en büyük cluster (tüm noktaları kapsayan) ve en altında en küçük cluster (her cluster tek noktadan oluşur) olacak şekilde iç içe clusterlar oluşmuş olur:



* Cluster'lar arası mesafe single link (en yakın iki nokta arası mesafe), complete link (en uzak iki nokta arası mesafe), average link (tüm nokta çiftleri arası mesafelerin ortalaması) veya centroidler arası mesafe baz alınarak hesaplanabilir

Divisive (Top Down) Clustering:

- En üstten başlayarak ikiye böle böle gider (Agglomerative'in tam tersi)

Distance Functions:

- Numerical attributeler için:
  - Euclidean (x=3, y=4) => distance = sqrt(3**2 + 4**2) = 5
  - Manhattan (x=3, y=4) => distance = 3+4 = 7
  - Squared euclidian: $dist(\mathbf{x}_i, \mathbf{x}_j) = (x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + ... + (x_{ir} - x_{jr})^2$
  - Chebychev distance: $dist(\mathbf{x}_i, \mathbf{x}_j) = \max(|x_{i1} - x_{j1}|, |x_{i2} - x_{j2}|, ..., |x_{ir} - x_{jr}|)$

- Binary attributeler için:

Data point *i* \ Data point *j*	1	0	
1	*a*	*b*	*a+b*
0	*c*	*d*	*c+d*
	a+c	*b+d*	*a+b+c+d*

  - Symmetric (eşit öneme sahip) binary attributes: (b+c)/(a+b+c+d)

$\mathbf{x}_1$	1	1	1	0	1	0	0
$\mathbf{x}_2$	0	1	1	0	0	1	0

$$dist(\mathbf{x}_i, \mathbf{x}_j) = \frac{2+1}{2+2+1+2} = \frac{3}{7} = 0.429$$

  - Asymmetric (farklı öneme sahip) binary attributes: (b+c)/(a+b+c) (Jaccard Coeff.) (Weight, varation eklenebilir)
- Nominal attributeler için:
  - dist(x,y) = (r-q)/r (r: attribute sayısı, q: eşleşen attributeler)
- Text documents için:
  - Cosine similiarity (daha sonra göreceğiz)

Evaluation Metrics of a Cluster:

- Compactness: intra-cluster noktaların yakınlığı
- Isolation: inter-cluster noktaların uzaklığı
- Entropy: cluster içindeki noktaların kararsızlığı, belirsiziliği
- Purity: cluster içinde noktaların dağılımının homojenliği

Cluster	Science	Sports	Politics		Entropy	Purity
1	250	20	10		0.589	0.893
2	20	180	80		1.198	0.643
3	30	100	210		1.257	0.617
Total	300	300	300		1.031	0.711